

## 5 Séquence numéro 5 : polymorphisme. 4h TP

### 5.1 TP - Chargement d'un niveau.



A l'heure actuelle, votre projet doit être capable d'afficher un donjon graphique avec le héros se déplaçant dedans, et une physique de collision élémentaire. Si ce n'est pas le cas, vous pouvez vous resynchroniser avec l'avancé du TP en utilisant la solution en ligne à l'adresse suivante : <https://github.com/antoineTauvelENSEA/Whatever2D/tree/SeanceTPNumero3>

#### 5.1.1 Chargement d'un niveau



A l'aide des classes `FileReader` et `BufferedReader`, codez un constructeur de `Dungeon` prenant en entrée une chaîne de caractère `fileName` et un `TileManager`.

Le constructeur lit une première fois le fichier, pour déterminer la taille du donjon et initialiser le tableau `map`.

Puis le constructeur lit une seconde fois le fichier pour remplir la `map`.

On doit ensuite modifier la méthode `respawnListOfThings` pour s'adapter au `TileSet` fixé. Par exemple, j'ai associé à W la tile (0,2) et au sol la tile (0,0).

**Si le fichier de niveau n'existe pas ou est corrompu, le constructeur appelle alors le constructeur par défaut avec des dimensions arbitraires.**

Testez le fonctionnement sans erreur et avec erreur.

Pour rendre l'ensemble plus attrayant, vous pouvez intégrer d'autre caractère selon l'orientation du mur, les angles etc.

#### 5.1.2 Modification du déplacement du héros (optionnel)

Le héros possède un déplacement avec une grande inertie. On peut coder son déplacement dans un timer qui ferait office de moteur de jeu conjointement au timer graphique. Cela donne le code suivant :

---

```

ActionListener animationTimer=new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        repaint();
        final int speed=10;
        if (hero.isWalking()){
            switch (hero.getOrientation()){
                case LEFT: hero.moveIfPossible(-speed,0,dungeon);
                           break;
                case RIGHT: hero.moveIfPossible(speed,0,dungeon);
                           break;
                case UP:    hero.moveIfPossible(0,-speed,dungeon);
                           break;
                case DOWN: hero.moveIfPossible(0,speed,dungeon);
                           break;
            }
        }
    }
};
Timer timer = new Timer(50,animationTimer);
timer.start();

```

---

Listing 10: Code du déplacement du héros



A l'aide de ce code et en modifiant l'implémentation de l'interface KeyListener, réimplantez le mouvement du héros.

### 5.1.3 Animation d'élément graphique

En reprenant le travail fait pour le héros, modifiez la méthode draw de la classe Animated Things. Elle va passer plusieurs image de manière à proposer une animation pour les torches. A vous de jouer pour la mettre en oeuvre.

En mettant des torches dans le donjon par le biais de la liste listOfThings, mettez en évidence le polymorphisme.